Self-Paced Balance Learning for Clinical Skin Disease Recognition

Jufeng Yang, Xiaoping Wu, Jie Liang, Xiaoxiao Sun, Ming-Ming Cheng, Paul L. Rosin and Liang Wang

***Abstract*—Class imbalance is a challenging problem in many classification tasks. It induces biased classification results for minority classes which contain less training samples than others. Most existing approaches aim to remedy the imbalanced number of instances among categories by re-sampling the majority and minority classes accordingly. However, the imbalanced level of difficulty of recognizing different categories is also crucial, especially for distinguishing samples with many classes. For example, in the task of clinical skin disease recognition, several rare diseases have a small number of training samples, but they are easy to diagnose because of their distinct visual properties. On the other hand, some common skin diseases, *e.g.*, eczema, are hard to recognize due to the lack of special symptoms. To address this problem, we propose a self-paced balance learning (SPBL) algorithm in this paper. Specifically, we introduce a comprehensive metric termed the *complexity of image category* which is a combination of both sample number and recognition difficulty. First, the complexity is initialized using the model of the first pace, where the pace indicates one iteration in the self-paced learning paradigm. We then assign each class a penalty weight which is larger for more complex categories and smaller for easier ones, after which the curriculum is reconstructed by rearranging the training samples. Consequently, the model can iteratively learn discriminative representations via balancing the complexity in each pace. Experimental results on the SD-198 and SD-260 benchmark datasets demonstrate that the proposed SPBL algorithm performs favorably against the state-of-the-art methods. We also demonstrate the effectiveness of the SPBL algorithm's generalization capacity on various tasks such as indoor scene image recognition, object classification, etc.**

***Index Terms*—Class imbalance, self-paced balance learning, clinical skin disease recognition, complexity level**

I. Introduction

THE number of training samples for each skin disease depends heavily on its incidence [1]–[3]. Actually, there are more than one thousand kinds of skin diseases, both common and uncommon, for which it is difficult to either collect or annotate a balanced dataset. Fig. 1 shows the histograms of image number distributions in two skin disease datasets, *i.e.*, SD-198 (top) [4] and SD-260 (bottom), where the images are captured by the digital camera or mobile phone, uploaded by patients, and labeled by doctor volunteers. In this figure, the blue bars reflect a large gap in the number of samples between common skin diseases, *e.g.*, solar elastosis (SE), allergic contact dermatitis (ACD), acne vulgaris (AV) and benign keratosis (BK), and uncommon skin diseases, *e.g.*, vitiligo (VI), stomatitis (ST), pilomatrixoma (PI) and histiocytosis X (HX). However, as shown by the red line, the recognition accuracy of each category is independent of the number of samples, indicating that the recognition difficulty is also imbalanced for the disease classes. According to empirical medical knowledge [5], some rare skin diseases, *e.g.*, ST and HX, have distinct characteristics and are easy to diagnose, while some common skin diseases, *e.g.*, ACD and BK, are difficult to recognize due to the lack of special symptoms.

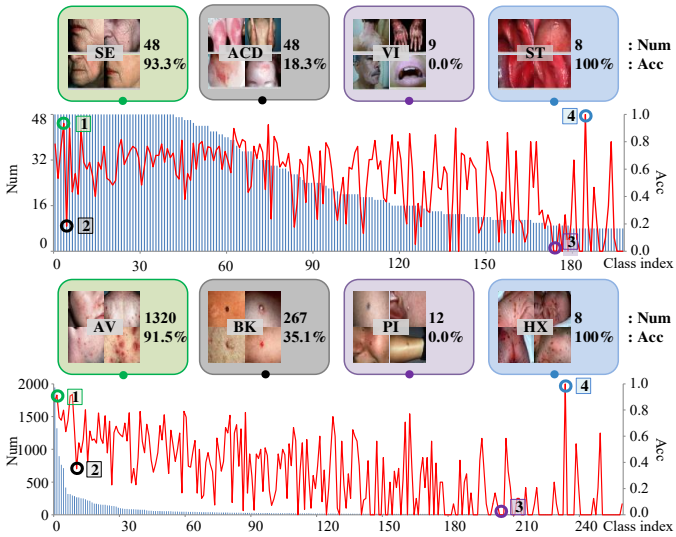


Fig. 1: Visualization of the class distribution in the SD-198 [4] (top) and SD-260 (bottom) datasets. The blue bars denote the number of training samples (Num) for each class, while the red line denotes the classification accuracy (Acc) of the raw ResNet50 [6] on the testing set. Each colored box visualizes a specific skin disease, *e.g.*, solar elastosis (SE), allergic contact dermatitis (ACD). The numbers in the boxes report the number of samples and the recognition accuracy, respectively.

However, most existing works on class imbalance problems focus only on the imbalanced distribution of sample numbers among different classes [7]–[9]. Such distribution indicates a large gap in the training numbers among categories [10]–[13], where there mainly exists three types of solutions, *i.e.*, the sampling-based [14]–[16], the cost-sensitive based [17]–

This manuscript is submitted to the special issue on Recent Advances in Theory, Methodology and Applications of Imbalanced Learning.

J. Yang, X. Wu, J. Liang, X. Sun and M.-M. Cheng are with College of Computer Science, Nankai University, Tianjin, 300350, China. Email: yangjufeng@nankai.edu.cn, xpwu95@163.com, liang27jie@163.com, sunxiaoxiaozrt@163.com, cmm@nankai.edu.cn

P.L. Rosin is with School of Computer Science and Informatics, Cardiff University, Wales, UK. Email: RosinPL@cardiff.ac.uk

L. Wang is with the National Laboratory of Pattern Recognition, CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. E-mail: wangliang@nlpr.ia.ac.cn

[19], and the ensemble-based methods [20], [20], [21]. Among them, the sampling-based ones attempt to balance the number of samples in the training dataset either by over-sampling the minority classes or under-sampling the majority ones. However, this re-sampling strategy may add redundant noisy data or lose the informative training samples. In comparison, the cost-sensitive based methods usually improve the classification sensitivity according to class-dependent costs when handling minority classes. Such costs are calculated by several heuristics based on prior knowledge, such as the imbalanced ratio of the sample numbers. Different from them, the ensemble-based methods construct a set of learning branches and then combine their decisions. Although the ensemble scheme has advantages over single methods, it relies heavily on the experimental tuning to properly combine the individual classifiers, which may result in unsatisfactory performance for practical applications.

In this paper, we address the class imbalance problem via a combined complexity metric termed the *complexity of image category* which synthesizes both the sample number and recognition difficulty of classes. We then design a self-paced balance learning (SPBL) framework inspired by the self-paced learning (SPL) paradigm [22], [23] to construct a dynamic program according to the updated complexity. Here, the SPL simulates the process of teaching a curriculum for students which arranges the samples from easy to difficult during training. It guides the learning procedure to avoid biased results towards the easily recognized categories (*e.g.*, those with large class sizes and small intra-class variation).

In addition, we use the iterative SPL scheme to arrange the learning process using the complexity of image categories. Specifically, we divide the learning process into K paces. Given $\{N_i\}_{i=1}^{C}$ training samples in C classes, we randomly select $\{N_i/K\}_{i=1}^{C}$ of them for each class in the first pace, while the others are used for evaluation. These $\sum_{i=1}^{C} N_i/K$ samples construct the first curriculum to train the initialized model. In the following paces, the mean loss of each category is calculated by the model derived from the last pace, which is used to measure the recognition difficulty of this category. Then, the complexity score of each image category is calculated based on a trade-off of both the class size and recognition difficulty. During training, a wrong prediction of any images in complex classes is assigned with high penalty weights to train a better classifier. Given the set of complexity scores, we reconstruct a new curriculum by selecting samples from the remaining training samples accordingly. Finally, we re-train the classifier with the updated penalty weights and curriculum, and also fine-tune the feature extractor on the current curriculum.

We validate the proposed framework on a clinical skin disease recognition task with a public dataset SD-198 [4] and a newly-collected one called SD-260. As shown in Fig. 1, the SD-198 dataset contains 198 categories of skin diseases, each of which has 10 to 60 images. However, according to the illustration by Sun *et al*. [4], the class distribution in the real applications might be more extreme than in this dataset, since they only preserve 60 samples for the classes which contain a large number of images and ignore those consisting of less than 10 images to avoid creating imbalanced class sizes. Consequently, in this paper, we collect an imbalanced skin disease dataset termed SD-260 according to the real distribution of class sizes reflected by the DermQuest[1] website, where the maximum class contains $2,432$ images and the minimum one contains 10 samples. Fig. 1 shows the class distribution of both challenging datasets, which show imbalanced distributions on both class size and recognition difficulty. We also extend our method to many alternative imbalanced tasks such as indoor scene image recognition and object classification. Extensive experiments on the evaluated datasets demonstrate the favorable performance of the proposed SPBL algorithm.

The contributions of this work are summarized as follows:

1) We propose the *complexity of image category* which alleviates the class imbalance considering both the class sizes and the recognition difficulties of each category.
2) We propose the *self-paced balance learning (SPBL)* algorithm to dynamically update those complexities, followed by attaching penalty weights and reconstructing a curriculum for discriminative representations.
3) To better evaluate the proposed SPBL method, we collect a new clinical skin disease dataset termed SD-260 which contains 260 classes of skin diseases and $20,600$ clinical images.

Experimental results on both the SD-198 and SD-260 datasets and several extended tasks demonstrate that the proposed SPBL algorithm outperforms the state-of-the-art methods. We will release all the code, data and learning models to the community.

The remaining part of this paper is organized as follows. In Section II, we briefly review the related works. In Section III, we illustrate the details of the proposed SPBL algorithm. Experimental results and analysis are then provided in Section IV. Section V concludes this paper.

II. Related Work

In this section, we briefly review the literature [24], [25] of class imbalance, self-paced learning, and clinical skin disease recognition tasks.

A. Class Imbalance

Deep learning technology recently attracts many researchers' attention on the object classification [26]–[28], detection [29]–[32], and other fields [24], [33]–[35], yet the balanced training data is scarce in practical applications. How to tackle class imbalance is an important issue in visual recognition tasks. Several excellent surveys concerning imbalanced learning field are published in the past decade. He and Garcia [36] propose a systematic review of the problem fundamental, detailed solutions, and the major performance evaluation metrics under the imbalanced learning scenario. More recently, [37], [38] analyses the intrinsic characteristics of the imbalanced data. Branco *et al*. [39] then focus on a more general issue of imbalanced predictive modeling. Overall, the existing solutions of the class imbalance can

[1] https://www.dermquest.com/

mainly be grouped into three categories: the sampling-based, cost-sensitive based, and ensemble-based methods.

1) Sampling-Based Methods: Sampling-based methods attempt to handle the class imbalance problem at the data level, *i.e.*, improving the data preprocessing technique. Specifically, these methods aim to balance the distribution of the original training set by over-sampling the minority classes [40]–[43], under-sampling the majority classes [7], [44], [45], or both.

The over-sampling approaches try to duplicate some instances or create new samples from existing minority classes. However, this data augmentation process might inherently produce information redundancy [36], [46]. To address this, SMOTE [47] is proposed to generate synthetic instances by linear interpolating the nearest positive neighbors of minority class instances.

In contrast, the under-sampling approaches attempt to remove instances from the majority classes before training the classifier. This sampling strategy, which is often preferred to over-sampling [44], is easy to implement and efficient. However, it may lose critical information, especially for small datasets.

2) Cost-Sensitive Based Methods: Instead of adjusting the distribution of imbalanced data through various instance manipulating strategies, the cost-sensitive based methods assign suitable cost parameters to penalize the misclassification situations at the classifier level [8], [27], [48], [49]. In particular, a heavier penalty factor is applied to the misclassification of the minority classes compared to the majority ones, which improves the sensitivity of classifier. Hence, it is important to design a cost matrix that reveals the penalty for misclassified instances from one class to another. For example, [48] and [49] preset the cost parameters using prior knowledge, although they can dynamically adjust and learn them during a training phase according to the imbalance ratio of one class relative to the other classes. In addition, Zhou and Liu [17] indicate that most research focuses on class-dependent costs [8], [27], [48], [49]. While there are only a few investigations on instance-dependent costs [50], [51], they are more appropriate for real-world applications.

3) Ensemble-Based Methods: Ensemble-based methods (*e.g.*, MOS-ELM [52]) usually construct a set of learning algorithms and then combine their decisions. Adapting either boosting (*e.g.*, AdaCost [53], RealBoost [54], and LogitBoost [55]) or bagging (*e.g.*, [56]) to use a sampling technique is a popular choice for class imbalance learning [57]. Specifically, [9], [58], [59] show that boosting ensembles perform better than the simplest approaches. In addition, [60] and [61] employ bagging to re-sample neighbor instances from minority classes. Besides, at the algorithm level, different cost-sensitive based boosting algorithms [62], [63] attempt to minimize the number of the high-cost errors and the total cost for improving accuracy and reduction in learning time for classification tasks. Furthermore, Wang *et al.* [64] propose an ensemble strategy that combines transfer learning and meta-learning to address the problem of long-tail recognition. Supported by empirical evaluations, all of them achieve favorable performance compared to using any single method.

B. Self-paced Learning

Self-paced learning is an important technique in the machine learning community [65], [66]. It simulates the cognitive system of human which at first learns an initialized and generalized model structure, followed by increasing the complexity to accomplish the task of learning comprehensive and technical knowledge. Among existing methods, the measurement of complexity scores of each class or sample is at the core of this problem. In addition, the updating of learning systems from easy to hard according to such complexities is also important.

Inspired by the regular learning pattern of humans, Bengio *et al.* [67] formalize a general training strategy termed curriculum learning (CL). CL aims to address a non-convex optimization problem by gradually progressing the training data with samples from easy to hard. Consequently, the critical issue in CL is to determine the order of such samples for the subsequent curriculum. However, it is difficult to define a clear distinction between easy and hard instances due to its ambiguous nature, especially for real-world and large-scale datasets.

To alleviate this problem, Kumar *et al.* [22] design a novel *self-paced learning* (SPL) paradigm with the same goal as the CL, where the training instances are presented in a meaningful order to facilitate the learning procedure. The SPL iteratively updates the importance parameter of instances rather than using fixed heuristic knowledge and trains a dynamical model. Meng *et al.* [23] further provide a theoretical understanding of SPL. Here, we briefly review the general form of the SPL paradigm.

Given a set of training data $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i$ and y_i denote the i-th ($i \in \{1, \cdots, n\}$) observed instance and the corresponding label, respectively. Let $L(y_i, g(\boldsymbol{x}_i, \boldsymbol{w}))$ represent the loss between the estimated label $g(\boldsymbol{x}_i, \boldsymbol{w})$ and its ground truth label y_i. The task of the SPL is to jointly learn the model parameter $\boldsymbol{w}$ and the latent weight variables $\boldsymbol{v} = [v_1, \cdots, v_n]^T$ by minimizing:

$$\min_{\boldsymbol{w}, \boldsymbol{v} \in [0,1]^n} \boldsymbol{E}(\boldsymbol{w}, \boldsymbol{v}, \lambda) = \sum_{i=1}^{n} \left(v_i L(y_i, g(\boldsymbol{x}_i, \boldsymbol{w})) + f(v_i, \lambda) \right), \quad (1)$$

where $f(v_i, \lambda)$ is called a self-paced regularizer (SP-regularizer [23]) with a monotonically increasing pace parameter λ. By controlling the loss value and the pace parameter, the model determines whether to include an instance into the learning process. Accordingly, the core of the SPL is to properly design the SP-regularizers, existing works include hard [22], linear [68] and mixture [69] SP-regularizers.

More recently, the theory of SPL has been successfully employed in various tasks, such as the multimedia search [68], matrix factorization [69], self-paced curriculum learning [70], co-saliency detection [71] and face identification [72], *etc*. However, these works rarely involve the issue of class imbalance which widely exists in real life, especially in medical imaging processing. Inspired by SPL, we propose a novel self-paced balance learning (SPBL) mechanism to solve the class imbalance problem. We propose to learn instances, ordered from easy to hard, while balancing the self-paced curriculum via penalty weight updating and curriculum reconstruction strategies.

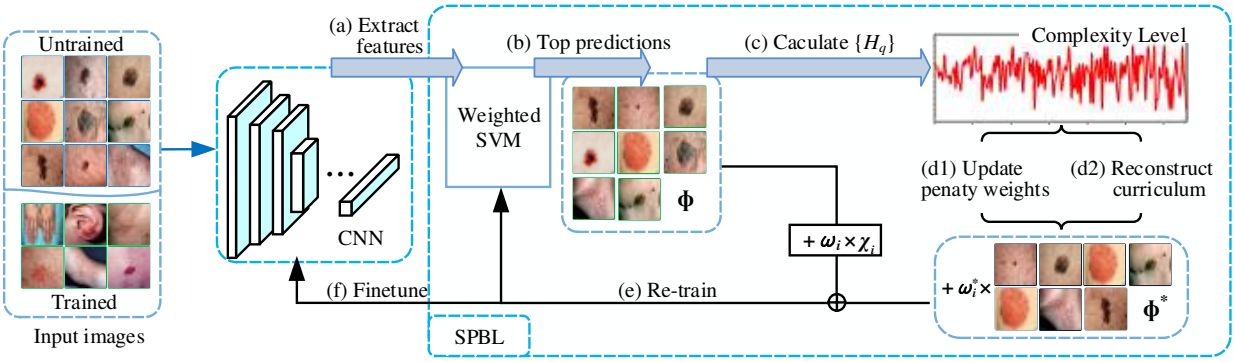


Fig. 2: Main steps of the proposed SPBL algorithm. It iteratively trains the weighted SVM classifier and updates the self-paced curriculum. The predictions with top scores form the initial curriculum $\mathbf{\Phi}$. During training, the algorithm calculates the distribution of class complexity level H, which combines both the class size and recognition difficulty. Based on that, we use a penalty weight updating strategy to calculate the class penalty weights $\boldsymbol{\omega}$, and use a curriculum reconstruction strategy to sample a balanced curriculum $\mathbf{\Phi} + \mathbf{\Phi}^*$ for training the SVM classifier in the next stage.

C. Clinical Skin Disease Recognition

A recent report in *Nature* [73] indicates that performing clinical skin disease recognition by image analysis is of major importance since skin disease is one of the most common diseases appearing in medicine, occurring widely in human life with significant ill effect. There are some related developments in this field, such as disease classification [74]–[76], lesion segmentation [77], detection and localization [78].

Previous works in skin disease recognition mostly focus on dermoscopic image processing [73], [76]–[78]. However, handling directly on clinical skin disease images is more economical, and getting the digital image from the portable electronic device (*e.g.*, mobile phone) is more convenient for patients who can then carry out self-diagnosis. Unfortunately, there are few open, large-scale standardized data sources [4] that are needed to develop deep learning technology in this field. Besides, researchers have to face the challenge that clinical imaging is easily affected by light intensity, camera angle, uncertain background, and other natural factors and interferences. Moreover, most current researches address binary skin disease recognition problems (*e.g.*, melanoma vs. non-melanoma skin cancer classification), while in practice clinical skin disease diagnosis needs to distinguish between large numbers of categories.

Apart from the above-mentioned issues, the class imbalance problem is also critical in clinical skin disease recognition task. Different diseases occur with differing frequencies, which may inherently cause datasets to have imbalanced training instances across classes. To the best of our knowledge, there have been no studies to incorporate SPL to tackle the class imbalance problem in skin disease recognition. We will introduce the proposed SPBL algorithm in the following section III in detail.

III. Methodology

We introduce the proposed SPBL framework in this section. First, we present the theoretical analysis and the formulation, as well as the choice of SP-regularizer [23], which is responsible for controlling the learning procedure and calculating the latent weight variables. Then, we introduce the definition and calculation of the complexity level of a class. Finally, we present two strategies for optimizing the SPBL based on the class complexity levels. Fig. 2 shows the pipeline of the proposed algorithm, in which the cost parameter updating, curriculum reconstructing, CNN model fine-tuning and classifier training are the main components of one pace.

A. Self-paced Balance Learning

In this work, we present the self-paced balance learning (SPBL) method to solve the class imbalance problem which computes the complexity of categories based on both the number of samples and the recognition difficulty of classes. The SPBL extends the self-paced learning paradigm (Eq. (1)) in two ways: 1) penalizing the classification errors with larger weights on the more complex categories; 2) reconstructing the curriculum for the following pace which re-balances the class distribution based on both the number of samples and the recognition difficulty. The optimization objective of the SPBL scheme is defined as:

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{v}\in[0,1]^n} \boldsymbol{E}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\omega},\lambda,\mathbf{\Phi}^*) = \\ \sum_{i=1}^{n} \omega_i \left(v_i L(y_i, g(\boldsymbol{x}_i,\boldsymbol{w})) + f(v_i,\lambda)\right) + \\ \sum_{j=n+1}^{m} \omega_j \left(L(y_j, g(\boldsymbol{x}_j,\boldsymbol{w}))\right) + \frac{1}{2}\|\boldsymbol{w}\|_2, \\ s.t.\ \{(\boldsymbol{x}_{n+1}, y_{n+1}), \cdots, (\boldsymbol{x}_m, y_m)\} \in \mathbf{\Phi}^*,\end{aligned} \tag{2}$$

where $\boldsymbol{v} = [v_1, \cdots, v_n]^T$ is the set of latent weight variables which controls the selection of training instances. In addition, $\boldsymbol{\omega}$ denotes the set of penalty weights which is harder on the misclassification of samples from more complex classes, and $L(y_i, g(\boldsymbol{x}_i,\boldsymbol{w}))$ computes the loss between predicted label $g(\boldsymbol{x}_i,\boldsymbol{w})$ and its ground truth label y_i. $f(v_i,\lambda)$ is the self-paced regularization term (SP-regularizer [23]) with an increasing pace parameter λ. $\mathbf{\Phi}^*$ denotes the reconstructed curriculum based on the original self-paced curriculum $\mathbf{\Phi}$ (details can be found in Section III-C). Moreover, n denotes the total number of training samples, while m denotes the number of extended samples copied from the last curriculum for those minority classes.

The SP-regularizer $f(v,\lambda)$ is designed to control the pace of the learning procedure and to regularize the latent weight variables. Several SP-regularizers have been constructed, including hard [22], linear [69] and mixture [68] forms. In this work, we use the typical hard SP-regularizer [22] as follows:

$$f(v_i,\lambda) = -\lambda v_i, \tag{3}$$

of which the closed-form solution $v^*(\lambda, L)$ is:

$$v^*(\lambda, L) = \begin{cases} 1, & \text{if } L < \lambda \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Here, the i-th instance will be added into the current curriculum $\mathbf{\Phi}$ if we have $L < \lambda$. During training, we optimize both the model parameter $\boldsymbol{w}$ and latent weight variables $\boldsymbol{v}$ in Eq. 2 by alternately optimizing one of them while fixing the other.

B. Complexity Level of Classes

We define the complexity level of a class in this section which is a trade-off between both the class size and recognition difficulty. We use the loss $L(y_i, g(\boldsymbol{x}_i, \boldsymbol{w}))$ to measure the recognition difficulty of $\boldsymbol{x}_i$ which is calculated based on the cross entropy loss function as follows:

$$L(y_i, g(\boldsymbol{x}_i, \boldsymbol{w})) = -\log p(y_i|\boldsymbol{x}_i). \tag{5}$$

Here, $p(y_i|\boldsymbol{x}_i)$ is the probability of correctly classifying the sample $\boldsymbol{x}_i$.

In the training stage, we divide the learning process into K paces based on the standard SPL paradigm. In the first pace, we randomly select $\frac{n}{K}$ training samples from each category to construct the first curriculum and train a model. We then calculate the recognition loss on the remaining $\frac{(K-1)n}{K}$ samples using this model and select the other $\frac{n}{K}$ with the smallest recognition losses to calculate the similar loss in the next pace using the newly-trained model. Given that, we define the recognition difficulty $l_{c_q^k}$ of a class c_q in the k-th pace, where $k \in \{1, \cdots, K\}$, $q \in \{1, \cdots, C\}$ and C is the number of categories. Specifically, we compute the average loss among the newly-selected training samples of each class to denote recognition difficulty of this class as follows:

$$l_{c_q^k} = -\frac{1}{|c_q^k|}\sum_{j=1}^{|c_q^k|} \log p(y_j|\boldsymbol{x_j}),\ x_j \in c_q^k, \tag{6}$$

where $|c_q^k|$ is the number of newly-selected samples from the remainder of the set of class c_q in curriculum $\mathbf{\Phi}^*$. Here, we have $\sum_{q=1}^{C}|c_q^k| = |\mathbf{\Phi}|^* - |\mathbf{\Phi}|$, where $|\mathbf{\Phi}|^* - |\mathbf{\Phi}|$ denotes the number of total newly-selected samples from the curriculum $\mathbf{\Phi}$ for the new $\mathbf{\Phi}^*$. Note we calculate the recognition difficulty based on the newly-selected data rather than the whole set of samples to simultaneously speed up the training process and precisely evaluate the difficulty with unseen samples of the model.

We then define the complexity level H_q^k of the class c_q in the k-th pace as follows:

$$\begin{aligned} H_q^k &= \frac{\exp(l_{c_q^k})}{|c_q^k|} \\ &= \frac{1}{|c_q^k|\prod_{j=1}^{|c_q^k|} p(y_j|\boldsymbol{x_j})^{\frac{1}{|c_q^k|}}}. \end{aligned} \tag{7}$$

For an arbitrary class, if the recognition difficulty is larger and the number of instances is smaller, then the complexity H_q^k is larger than others. The set of complexity levels, *i.e.*, $\{H_q^k\}_{q=1}^C$, is used to update the penalty weights of the k-th pace, which is explained in Section III-C1.

C. Alternative Optimization of SPBL

Based on the complexity levels of classes, we alternately update the penalty weights, reconstruct the curriculum and retrain the model.

1) Penalty Weight Updating: We illustrate the calculation of the penalty weight $\boldsymbol{\omega}$ in Eq. 2 in this subsection. First, we define a cost matrix $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{C\times C}$ and denote by $\boldsymbol{\mathcal{C}}_{ij}$ as the misclassification cost where the samples of class c_i are predicted as c_j. The cost matrix $\boldsymbol{\mathcal{C}}$ satisfies the following conditions: 1) $\boldsymbol{\mathcal{C}}_{ii} = 0$; 2) $1 \leqslant \boldsymbol{\mathcal{C}}_{ii} \leqslant \alpha$ for $i \neq j$ where α denotes a predetermined upper limit of the cost; and 3)there exists at least one pair of classes where $\boldsymbol{\mathcal{C}}_{ij} = 1$. We then follow the definition of [79] to represent the misclassification $cost(i)$ of class c_i:

$$\boldsymbol{\mathcal{C}}_i = \sum_{j=1}^{C} \boldsymbol{\mathcal{C}}_{ij}. \tag{8}$$

For an arbitrary pair of classes c_a and c_b, we have $\boldsymbol{\mathcal{C}}_a \leqslant \boldsymbol{\mathcal{C}}_b$ if $H_a \leqslant H_b$.

We then define the penalty weight $\boldsymbol{\omega}_i$ of class c_i as follows:

$$\omega_i = \boldsymbol{\mathcal{C}}_i \frac{\sum_{j=1}^{C}\frac{1}{H_j}}{\sum_{j=1}^{C}\boldsymbol{\mathcal{C}}_j\frac{1}{H_j}}, \tag{9}$$

where we have $\sum_{i=1}^{C}\omega_i\frac{1}{H_i} = \sum_{i=1}^{C}\frac{1}{H_i}$. Moreover, the set of penalty weights $\boldsymbol{\omega}$ are normalized by

$$\boldsymbol{\omega}^* = \frac{\boldsymbol{\omega}}{\min(\boldsymbol{\omega})} \tag{10}$$

where we have $\min(\boldsymbol{\omega}^*) = 1$ since the easiest class does not need to be penalized.

2) Curriculum Reconstruction: The self-paced learning algorithm progressively trains the model using samples from easy to hard. However, this regime only benefits the imbalanced recognition difficulty problem yet overlooks the imbalance size among the classes. To overcome this weakness, the proposed SPBL re-balances the class distribution of the curriculum $\mathbf{\Phi}$ via a novel curriculum reconstruction strategy:

$$|c_i^*| = \underset{|c_i|}{\operatorname{argmin}}\left(\frac{\exp(l_i)}{|c_i|} - \frac{\sum_{j=1}^{C}\exp(l_j)}{\sum_{j=1}^{C}|c_j|}\right), \tag{11}$$

where $|c_i^*|$ indicates the final number of training samples of class c_i in the current pace.

Algorithm 1 Curriculum Reconstruction Algorithm

Input: Original curriculum $\mathbf{\Phi}$, penalty weight $\boldsymbol{\omega}$
Output: Reconstructed curriculum $\mathbf{\Phi}^*$, updated penalty weight $\boldsymbol{\omega}^*$

Calculate the recognition difficulty of each class via Eq. 6;
Calculate the average complexity level among all classes via $\frac{\sum_{j=1}^{C}\exp(l_j)}{\sum_{j=1}^{C}|c_j|}$;
for $i = \{1, \cdots, C\}$ **do**
  Calculate the final number of instances $|c_i^*|$ of the class c_i in $\mathbf{\Phi}^*$ via Eq. 11;
  if $|c_i^*| > |c_i|$ **then**
    Copy $|c_i^*| - |c_i|$ instances of class c_i with top losses to $\mathbf{\Phi}^*$;
    Set $\omega_i^* = \omega_i$;
  else
    Remove $|c_i| - |c_i^*|$ instances of class c_i with top losses;
    Set $\omega_i^* = 0$;
  end if
end for
return $\{\mathbf{\Phi}^*, \boldsymbol{\omega}^*\}$

To balance the complexity level among classes, we dynamically assign the number of instances for each class which are added to the curriculum based on Eq. 11. If we have $|c_i^*| > |c_i|$, then $|c_i^*| - |c_i|$ instances of the class c_i are added into the reconstructed curriculum $\mathbf{\Phi}^*$ using over-sampling strategy. Specifically, we copy the samples with top $|c_i^*| - |c_i|$ losses to over-sample this category. Meanwhile, we set the weight parameter $\omega_i^* = \omega_i$. On the contrary, if $|c_i^*| \leqslant |c_i|$, we remove $|c_i - c_i^*|$ instances of the class c_i which have the top losses to under-sample this category. We set $\omega_i^* = 0$ in this case. The detailed process of the curriculum reconstruction is summarized in Algorithm 1.

3) CNN Model Tuning & SVM Classifier Training: At the beginning of training SPBL, the curriculum was initialized by a random set which contains $\frac{1}{K}$ of the entire training set. We fine-tune a pre-trained CNN model on this set to extract an initial feature representation for $\{\boldsymbol{x}_i\}_{i=1}^n$. After the updating of the learning pace, the curriculum size is gradually extended, where the model is fed with more training samples and learns more potential patterns from them. The feature extraction model, the classifier and the curriculum are then alternately updated in the training procedure.

To update the classifier, we fix $\{\{\boldsymbol{x}_i\}_{i=1}^u, \{y_i\}_{i=1}^u, \boldsymbol{v}, \boldsymbol{\omega}, \mathbf{\Phi}^*\}$ in both the CNN model and the curriculum and update the parameters $\boldsymbol{w}$ as follows:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}\in[0,1]^n}{\operatorname{argmin}} \sum_{i=1}^{n} \omega_i v_i L_i + \sum_{j=n+1}^{m} \omega_j L_j + \frac{1}{2}\|\boldsymbol{w}\|_2, \quad (12)$$
$$s.t.\ \{(\boldsymbol{x}_{n+1}, y_{n+1}), \cdots, (\boldsymbol{x}_m, y_m)\} \in \mathbf{\Phi}^*$$

where $L_i = (y_i, g(\boldsymbol{x}_i, \boldsymbol{w}))$ denotes the loss function. There are several classification algorithms adapted to our model. We employ a weight SVM in this paper as the classifier, where we assign the penalty weight in Eq. 10 to each class before classification.

Algorithm 2 Self-paced Balance Learning Algorithm

Input: Training dataset $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$
Output: Classifier parameter $\boldsymbol{w}$

Initialize the model with a pre-trained CNN and classifier parameters $\boldsymbol{w}$;
Initialize the SP-regularizer f, latent weight variables $\boldsymbol{v}$ and pace parameter λ;
Predetermine the initial curriculum $\mathbf{\Phi}$;
repeat
  Update $\boldsymbol{w}$ via Eq. 12;
  Update $\boldsymbol{v}$ via Eq. 4, and then get the curriculum $\mathbf{\Phi}$;
  Update the complexity level of each class via Eq. 7;
  Update penalty weight parameter $\boldsymbol{\omega}$ via Eq. 10;
  Update reconstructed curriculum $\mathbf{\Phi}^*$ and weight $\boldsymbol{\omega}^*$ via Algorithm 1;
  Tune the CNN model and extract features;
  In every T epochs:
    Augment λ;
until Model converge
return $\boldsymbol{w}$

4) Pace Parameter Updating: The pace parameter λ controls the number of training instances to be selected in the SPBL (before reconstruction of the curriculum), and it monotonically increases during the entire training procedure. Apparently, more difficult instances are included in the curriculum along with the processing of paces. As a result, we terminate the updating of the pace parameters when we get stable evaluation performance. Such termination is required because a difficult instance always has a larger loss, which may result in a negative impact on the system performance since the instance could even belong to noisy data with incorrect labels. To describe this, we refer to [23] to define a threshold λ_a on the losses where the pace parameter λ_a allows a instances to be added into the curriculum $\mathbf{\Phi}^*$, *i.e.*, there are a instances with a smaller loss than the pace parameter λ_a. Note in the early learning paces, most of the instances have a relatively small loss. Therefore, a small increase of the pace parameter λ will lead to a lot of untrained instances being added to the curriculum $\mathbf{\Phi}$.

5) Model Convergence: The entire alternate optimization process of the SPBL strategy is summarized in Algorithm 2. After initializing the parameters, the algorithm alternately updates one module while fixing the others, including the classifier parameters $\boldsymbol{w}$, the curriculum $\mathbf{\Phi}$ and the set of model parameters. Thus, the original overall optimization problem of Eq. 2 can be grouped into two sub-optimization problems, *i.e.*, the optimization of both the SPBL and the classifier. Note that at the beginning of the learning stage, the model is unstable like other typical SPL algorithms. While the size of the curriculum is increased along with the learning progresses, the model is trained with more patterns which leads to more robust and discriminative features extracted from the CNN model. Under the alternating optimization of the parameters, the objective function can decrease to an optimal

value iteratively. Thus the SPBL model becomes increasingly stable and finally achieves convergence.

IV. Experiments

In this section, we experimentally demonstrate the effectiveness of the proposed SPBL. Firstly, we introduce two benchmark datasets, *i.e.*, the SD-198 [4] and the SD-260 datasets, in which the samples among classes are imbalanced in terms of both the class size and the difficulty of recognition. Then, we illustrate the experimental settings including the model parameters and various evaluation metrics used for class imbalance learning. After that, we empirically evaluate and analyze the proposed SPBL algorithm on the two imbalanced datasets, and finally present the experimental results with comparison to the state-of-the-art methods. We also extend the proposed SPBL to several other tasks.

A. Datasets

The imbalanced problem in real-world applications is due to not only the imbalanced distribution of class sizes but also the recognition difficulty. Actually, both imbalanced problems are revealed in the clinical skin disease recognition task. Therefore, we mainly evaluate the proposed SPBL method on the SD-198 [4] and SD-260 datasets in this paper. These two datasets can be downloaded publicly [2]. We also extend the SPBL method on several other datasets including MIT-67 [80], Caltech-101 [81], MNIST [82] and MLC [83] datasets.

1) SD-198 Dataset: The SD-198 [4] dataset focuses on automatic skin disease recognition and diagnosis problem. It contains 198 categories of skin diseases and $6,584$ clinical images. Images in this dataset cover a lot of situations for patients such as gender (male, female), age (child, adult, old), disease site (head, nails, hand, feet), color of skin (white, black, brown, yellow), and different periods of lesions (early, middle, late). The images contain variations in color, exposure, illumination, and scale. These images were collected using digital cameras and mobile phones, uploaded by patients to the dermatology Dermquest website, and annotated by professional dermatologists.

2) SD-260 Dataset: When collecting the SD-198 dataset, the authors manually control the class size distribution by preserving $10-60$ images for each category [4]. As shown in Fig. 1, the SD-198 has a medium imbalance ratio [84] where the ratio of the largest category to the smallest one is about 6. This ratio is extremely different in real life where common and uncommon skin diseases have substantially different incidences. In this paper, we contribute a new skin disease dataset with a high imbalance ratio (larger than 243), named the SD-260 dataset. We collect the SD-260 from the same source as the SD-198, yet we only eliminate the class with less than 10 samples and preserve all other classes as well as all the available images of these diseases. Finally, it consists of 260 diseases and $20,600$ images, in which the maximum class has $2,432$ samples and the minimum one has 10. The increase of category number, the diversity among classes and the imbalance degree further leads to a more challenging dataset in the recognition task compared to the SD-198 dataset.

3) Extended Tasks: We also extend our proposed method to other tasks such as scene classification (MIT-67 [80]), object classification (Caltech-101 [81]), handwritten digit classification (MNIST [82]) and coral classification (MLC [83]). The MIT-67 [80] dataset contains $15,620$ images. The image numbers of 67 categories of the indoor scene vary between 101 and 738. The Caltech-101 [81] dataset contains $9,144$ images belongs to 102 categories (101 objects $+$ background). The image number for each category varies between 31 and 800. The MNIST [82] dataset consists of $70,000$ images and 10 categories of digits. Each category contains $7,000$ images. The MLC [83] dataset consists of $2,055$ images which are divided into three sets according to collection time ($2008-10$). Each image has roughly 200 point annotations belonging to 9 categories. The labelled points for each category approximately vary between $2,622$ and $196,910$.

B. Experimental Settings

1) Training/Testing Set Partition: We divide both the SD-198 and SD-260 datasets by randomly splitting each category into training and testing sets with $8:2$ samples. Specifically, we select $5,268$ images for training and the remaining $1,316$ images for testing in SD-198 and $16,480$ images vs. $4,120$ images in SD-260. Note the proportion between two different classes in the testing set is the same as in the training set as shown in Fig. 1. We follow the training/test split protocols from [27] in the extended tasks. We use the $6:4$ training/test split for the MIT-67 and Caltech-101 datasets and the $6:1$ for the MNIST dataset. In addition, for the MIT-67, Caltech-101, and MNIST datasets, we reduce the image number of odd classes to 10% in training set to unbalance training distribution. As for the MLC dataset, we train on the data of 2008 year and test on the data of 2009 year.

2) Network Parameters & Implementation Details: We use the raw ResNet-50 [6] which is pre-trained on ImageNet [85] as the backbone of the CNN architecture. We then fine-tune the network on the SD-198 and the SD-260 datasets, respectively. The learning rate is initialized to be 0.01 and decays by 0.1 in every 40 epochs. We use the Stochastic Gradient Descent (SGD) with momentum as the optimizer. The mini-batch size is set to be 64 and the momentum equals to 0.9. The weight decay parameter in the ℓ_2-regularization term is set to be 0.0005. The input RGB image size is fixed to a square of $224\times224\times3$ pixels. We implement the SPBL method using the open framework PyTorch, and run it on an Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz, 32 GB RAM, and an NVIDIA GeForce GTX TITAN X GPU with 12 GB VRAM. The code and pre-trained models are available online [3].

3) Evaluation Metrics: To avoid a compromise evaluation of misclassification among the minority and majority classes in class imbalance problem, we comprehensively measure the performance of the classifier on both the *precision* and *recall* using the following metrics: F-measure, G-mean [86] and M_{AUC} [87]. Assume n_{ij} is the number of samples in the class

[2]http://cv.nankai.edu.cn/projects/sd-198/

[3]https://github.com/xpwu95/SPBL_Pytorch

c_i that are classified as class c_j. Then the precision P_i and recall R_i of class c_i can be defined as:

$$P_i = \frac{n_{ii}}{\sum_{j=1}^{C} n_{ji}} \quad \text{and} \quad R_i = \frac{n_{ii}}{\sum_{j=1}^{C} n_{ij}}, \tag{13}$$

where C is the number of classes. The average precision and recall can be defined as:

$$\text{Precision} = \frac{1}{C}\sum_{i=1}^{C} P_i \quad \text{and} \quad \text{Recall} = \frac{1}{C}\sum_{i=1}^{C} R_i. \tag{14}$$

Neither of them can effectively represent the performance of classifier independently. The F-measure combines the precision and the recall as a trade-off with the choice that the factor $\beta = 1.0$ (F1) indicates recall and precision are equally important:

$$F\text{-measure} = \frac{1}{C}\sum_{i=1}^{C} \frac{\left(1+\beta^2\right) P_i R_i}{\beta^2 P_i + R_i}. \tag{15}$$

The G-mean evaluates the average sensitivity of all classes, and especially reflects the degree of bias in minority classes, which is defined as:

$$G\text{-mean} = \left(\prod_{i=1}^{C} R_i\right)^{\frac{1}{C}}. \tag{16}$$

As for the area under the curve (AUC) metric in the classification problem, we follow the micro average scheme M_{AUC} of the definition as in [7]. Similar to the form of F-measure and G-mean, it integrates the weighted average of all labels:

$$M_{\text{AUC}} = \frac{2M_{\text{P}}M_{\text{R}}}{M_{\text{P}} + M_{\text{R}}}, \tag{17}$$

where the micro average precision M_{P} and the recall M_{R} are defined as:

$$M_{\text{P}} = \frac{\sum_{i=1}^{C} n_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C} n_{ji}} \quad \text{and} \quad M_{\text{R}} = \frac{\sum_{i=1}^{C} n_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C} n_{ij}}. \tag{18}$$

C. Parameters

In this section, we discuss the setting of parameters of the proposed SPBL algorithm. We experimentally analyze the selection of the number of paces K and the pace parameter λ. In the SPBL algorithm, we keep the step-size of $\frac{n}{K}$ instances to expand the curriculum capacity in each paced learning procedure, where n denotes the number of instances in the total training set. We evaluate the SPBL performance under different settings of K from 1 to 7 under different performance metrics. As illustrated in Fig. 3, with the increase of K, the model will perform better within a certain interval. After comprehensively considering the trade-off between model complexity and performance, we set the total iteration number $K = 5$, and we monotonically augment the pace value to $\lambda_i \frac{n}{5}$ at the i-th pace of SPBL.

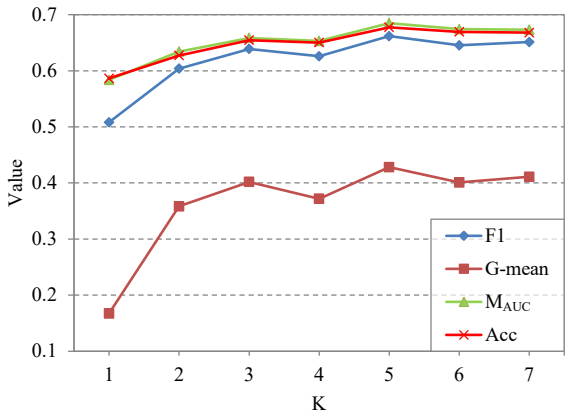


Fig. 3: Classification performance of the proposed SPBL method with different total number of paces (K). Here, "value" indicates the results of $F1$, G-mean, M_{AUC} and Acc (accuracy) on the SD-198 dataset. Accordingly, we set $K = 5$ in the rest of the experiments, *i.e.*, we conduct 5 paces for each experiment.

D. Ablation Study

We conduct a set of ablation experiments in this section to validate the effectiveness of each module of the proposed SPBL algorithm. Specifically, we evaluate the baseline of the self-paced learning (SPL) and two extended components, including penalty weight updating (PWU) and curriculum reconstruction (CR) strategies. We employ the ResNet-50 [6] as the deep feature extractor and the SVM as the classifier. Table I reports the experimental results.

1) Introducing SPL: We first evaluate the performance of self-paced learning [22], [23] which is introduced to address the class imbalance problem. As shown in Table I, the experimental results of SPL (second row of each dataset) on both imbalanced datasets demonstrate an improvement compared to the baseline method (using deep features to train SVM directly without any other processing, the first row of each dataset). Note when comparing the value of F1-measure, the SPL method leads to a big improvement of about 7%, which is mainly due to the incremental knowledge from hard instances and the effective learning process from easy to hard. The performance on the G-mean metric also shows a substantial increase of the SPL against the baseline method although both methods perform unsatisfactorily. However, there still exists a considerable gap between the G-mean and the accuracy. This reflects the fact that although SPL improves model performance over baseline, it still learns an insufficient representation and thus fails to handle the class imbalance adequately. For example, SPL cannot properly address the imbalanced situation in which one class not only has a few instances but is hard to learn.

2) Joint SPL and PWU Strategy: We then evaluate the effectiveness of the penalty weights updating (PWU) module in the SPBL architecture. As shown in Table I, adding the PWU module by setting the penalty parameter of the error term produces an improved accuracy of 3.7% on the SD-198 dataset. This is mainly because that the PWU intentionally

TABLE I: Ablation experiments on both the SD-198 and SD-260 datasets verifying the effectiveness of different modules of the proposed method. Each entry in this table is composed of the mean and variance of the corresponding performance derived by cross-validation.

Dataset	Method	F1	G-mean	M_{AUC}	Acc
SD-198	SVM	50.8±2.5	16.7±3.1	58.4±2.3	58.7±2.2
	SPL	57.8±2.6	27.5±1.7	63.1±2.9	62.2±3.1
	SPL+NPWU	61.1±1.9	34.5±2.9	64.2±2.0	63.6±1.9
	SPL+DPWU	58.3±2.7	31.7±3.3	63.5±2.4	62.9±2.1
	SPL+PWU	63.7±2.2	40.2±2.6	66.4±2.1	65.9±2.0
	SPL+CR	63.4±2.0	39.9±2.7	65.8±2.0	65.1±1.9
	SPBL	**66.2**±1.6	**42.8**±4.0	**68.5**±1.6	**67.8**±1.8
SD-260	SVM	33.6±1.0	4.2±0.3	59.2±0.6	60.9±5.8
	SPL	39.4±0.7	9.8±0.8	61.0±0.8	61.1±1.0
	SPL+NPWU	45.0±0.9	13.3±1.3	61.9±0.9	62.2±0.9
	SPL+DPWU	42.1±0.8	11.9±1.5	61.7±0.9	62.0±0.8
	SPL+PWU	48.2±1.0	15.5±1.3	63.0±0.8	63.6±0.8
	SPL+CR	48.4±0.9	15.9±1.1	62.7±0.8	63.3±0.7
	SPBL	**51.0**±0.9	**19.6**±1.1	**64.8**±1.2	**65.1**±0.8

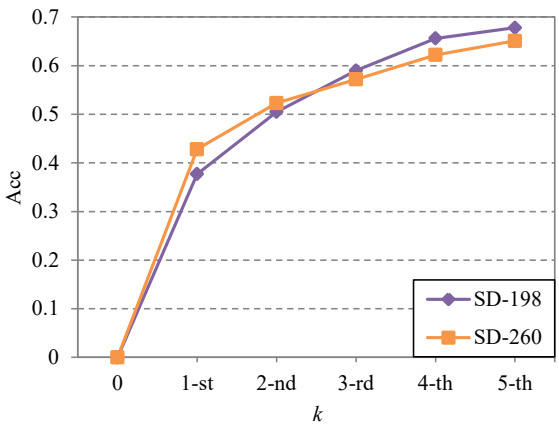


Fig. 4: Iterative performance along with paces when training the proposed SPBL algorithm on both the SD-198 (purple line) and SD-260 (brown line) datasets. Here, K indicates the total number of paces and k refers to one step. Note the classification accuracy is increased along with the increasing paces, while the result of the last pace outperforms the baseline method without self-paced learning strategy.

biases the learning among classes with higher complexity level, which forces the classifier to pay more attention to the more complex classes. Furthermore, in the PWU strategy, we alternatively replace the measurement of complexity level with the number of samples in class (NPWU) and recognition difficulty (DPWU), which reflect the individual effect of both class size and recognition difficulty. As shown in the table, the PWU strategy outperforms both alternatives under all evaluation metrics, which confirms the effectiveness of combining both the class size and recognition difficulty to measure the complexity level.

3) Joint SPL and Curriculum Reconstruction: We also explore the benefit of the curriculum reconstruction (CR) scheme on the data level. For a fair comparison, we fix the penalty weight of each class to be 1 in this experiment. As shown in Table I, the model with SPL and CR (SPL+CR) achieves similar performance as the model with PWU, both of which show a large improvement against the raw SPL method. The curriculum reconstruction strategy re-balances the class distribution of the curriculum from each self-paced learning procedure by over-sampling classes with a higher loss but fewer instances, and under-sampling classes with the lower loss but more instances. This fits the learning pattern of humans, *e.g.*, sometimes when we meet knowledge that is hard to learn, we need some easier cases to learn before. By emphasizing the importance of complex instances and weakening the redundant easy ones, the model incrementally learns and considers both the class size and difficulty from a balanced self-paced curriculum.

4) The Proposed SPBL: Finally, we integrate both the penalty weight updating and the curriculum reconstruction strategies and propose the SPBL method. Specifically, we first measure the class complexity level based on the original curriculum, then we use the complexity information to design the penalty weights and reconstruct the curriculum for each class. After that, we re-train the SVM classifier with the updated curriculum and weights. As shown in Fig. 4, the model achieves better performance in each step when using the self-paced learning procedure. Table I also demonstrate that the combination of PWU and CR strategies, *i.e.*, the SPBL method, outperforms others under all metrics.

E. Comparison with State-of-the-Art Methods

In this section, we compare our SPBL approach against the state-of-the-art methods on the SD-198 and SD-260 datasets and several other tasks.

1) Comparative Methods: All compared methods can be grouped into the four series as follows:

(i) Sampling-based methods: The sampling-based methods usually change the distribution of class sizes using re-sampling techniques, including under-sampling (*e.g.*, Random Under-Sampling **RUS**, Instance Hardness Threshold **IHT** [89], and **NearMiss-2** [88]) and over-sampling methods (*e.g.*, **ADASYN** [14], **SMOTE** [47] and Borderline-SMOTE **B-SMOTE** [90]). Among them, the **RUS** randomly removes samples to get a balanced class distribution. The **IHT** filters the datasets through a priori instance hardness information and integrates this knowledge into the training process to alleviate the effects of class overlap. The **NearMiss-2** chooses negative training samples by applying the k nearest neighbor approach. The **ADASYN** generates synthetic data for minority class samples according to their difficulty level in learning. The **SMOTE** operates in the feature space, and creates synthetic minority class instances by combining the sample under the observation with its nearest neighbor. The textbfB-SMOTE, unlike the **SMOTE**, only over-samples the minority instances near a decision boundary.

(ii) Cost-sensitive based methods: This kind of method penalizes the misclassification among classes via the cost factor of the classifier. The **Rescale**$_{new}$ [17] addresses the

TABLE II: Comparison to the state-of-the-art imbalanced learning methods on both the SD-198 and SD-260 datasets under different evaluation metrics. Each entry in this table is composed of the mean and variance of the corresponding performance derived by cross-validation.

Methods	SD-198						SD-260					
	Precision	Recall	F1	G-mean	M_{AUC}	Acc	Precision	Recall	F1	G-mean	M_{AUC}	Acc
RUS	58.1±1.6	55.6±1.5	53.1±1.0	31.2±1.1	57.3±1.4	54.6±1.4	36.7±1.2	42.0±1.1	35.2±0.9	15.2±1.9	49.4±1.3	45.3±1.0
NearMiss-2 [88]	58.0±1.4	57.4±1.8	54.8±1.4	34.1±3.6	58.3±1.5	57.0±1.6	30.9±1.3	42.2±1.4	31.4±1.1	15.7±1.9	43.6±1.5	36.3±1.8
IHT [89]	55.9±1.5	49.5±2.1	47.5±1.9	18.8±2.7	54.1±1.9	49.7±1.8	39.3±0.5	36.6±0.7	32.0±0.5	8.8±0.4	47.8±1.1	43.4±1.2
ADASYN [14]	64.4±1.7	63.4±1.9	61.7±1.6	41.0±2.8	64.9±1.8	64.1±1.8	55.6±0.8	47.9±0.1	49.4±0.3	18.5±0.5	63.8±0.4	64.3±0.3
SMOTE [47]	64.3±1.0	63.0±1.2	61.4±1.1	40.8±1.8	64.3±1.1	63.4±1.1	55.5±1.3	47.5±0.9	49.1±0.9	18.4±1.1	63.7±0.4	64.2±0.2
B-SMOTE [90]	63.1±0.8	60.9±1.6	59.7±1.3	39.3±2.5	63.1±1.7	62.2±1.8	55.6±1.1	47.1±0.8	48.9±0.8	17.7±1.2	63.4±0.4	64.1±0.2
Rescale$_{new}$ [17]	59.7±3.6	55.1±4.2	54.3±4.0	24.4±5.4	60.1±3.0	59.3±3.1	46.1±3.1	37.3±3.0	38.8±3.1	7.2±1.9	60.4±1.3	61.6±0.9
CSNN [48]	58.3±2.2	52.0±2.4	52.3±2.6	19.1±3.6	59.4±2.2	59.4±2.1	43.4±0.9	31.5±1.1	34.1±1.0	4.3±0.2	59.5±0.6	61.2±0.6
ENN [91]	64.7±2.0	59.0±2.1	59.3±2.1	34.5±5.3	63.0±2.0	61.3±1.9	52.6±1.7	46.9±1.4	45.9±1.4	**21.9**±1.3	60.2±0.6	55.2±1.6
SMOTEBoost [92]	61.5±2.1	58.7±4.7	57.2±3.5	32.7±7.6	61.8±3.0	60.7±2.7	41.8±1.8	39.3±1.0	38.4±1.2	7.9±0.4	58.2±0.5	60.2±0.5
RUSBoost [93]	56.3±1.7	53.1±1.9	52.3±1.8	19.1±1.3	59.3±2.0	59.5±2.0	39.8±1.2	38.3±0.7	36.7±0.8	7.5±0.5	57.7±0.6	57.8±0.5
SVM	56.6±2.0	50.8±2.4	50.8±2.5	16.7±3.1	58.4±2.3	58.7±2.2	42.6±1.3	31.0±0.9	33.6±1.0	4.2±0.3	59.2±0.6	60.9±5.8
SPBL	**71.4**±1.7	**65.7**±1.6	**66.2**±1.6	**42.8**±4.0	**68.5**±1.6	**67.8**±1.8	**59.9**±1.6	**48.2**±1.1	**51.0**±0.9	19.6±1.1	**64.8**±1.2	**65.1**±0.8

cost-sensitive learning by rescaling the classes using the cost information. The **CSNN** [48] trains cost-sensitive neural networks with a set of algorithms, in which threshold-moving is the best one and we compared against it in this work. The **ENN** [91] extends the nearest neighbor method to learn an unequal distribution, considering the relative nearest neighbor relationships between samples.

(iii) Ensemble-based methods: These methods usually employ several learning algorithms and combine their decisions. The **SMOTEBoost** [92] indirectly changes the updating weights of misclassified instances based on the combination of SMOTE and boosting learning. The **RUSBoost** [93] is another algorithm that combines boosting and data sampling, but is simpler and faster than the **SMOTEBoost**.

(iv) We also compare against two state-of-the-art methods, *i.e.*, [4] and [74], which are the typical solutions to address the class imbalance on clinical skin disease recognition problems. Table II and Table III show the comparisons of clinical skin disease recognition performance under six metrics including precision, recall, F1, G-mean, M_{AUC} and accuracy. Note the performance of comparative methods is not good on both datasets if we simply adopt the default hyper-parameters given in the original paper. In this paper, we tune the parameters of these methods and report the best result we got.

The comparison results against the state-of-the-art methods on the two datasets are reported in both Table II and Table III. The results from different strategies are grouped into different blocks of rows. For a fair comparison, we employ the same deep features derived from the same raw ResNet-50 model at the beginning step for all comparative methods and use the one-vs-rest scheme SVM with same parameter settings as the estimator.

Apparently, the original deep features combined with SVM estimator have poor performance as shown in the second last row of Table II. The results on the G-mean metric is especially worse than most compared methods. The G-mean calculates the geometric mean of the accuracies of every class, which means that the poor accuracy of even one class will lead to a poor G-mean value. Hence the result indicates that several classes are entirely unrecognized by the classifier, meaning there exists a massive imbalanced problem on both datasets.

2) Comparison to Sampling-Based Methods: As shown in Table II, the SPBL outperforms the under-sampling based methods on the SD-198 dataset with more discriminative representations and classifiers. However, the under-sampling methods, *e.g.*, NearMiss-2 and IHT, ensure that each class retains an approximate number of instances compared to the minority class, which may also cause the issue of few-shot learning. Moreover, the RUS performs better than IHT according to most metrics since the last method loses some useful information after removing instances. This weakness especially appears in the non-binary classification task with datasets that have a great disparity in the sizes of the majority and minority classes. In contrast, the SPBL dynamically removes instances with relatively simple information in each self-paced curriculum, which is demonstrated a positive effect on the classifier.

The SPBL also outperforms the over-sampling methods on all evaluation metrics. This is because that the SPBL focuses not only on the smaller classes but on the classes that are hard to classify no matter how many instances they have during training. In contrast, the sampling-based methods cannot process the imbalance of some classes since they only focus on the number of class, *e.g.*, that have a large number of instances and a high recognition difficulty. The experiments on the SD-260 dataset show similar results.

3) Comparison to Cost-sensitive Based Methods: We can observe from Table II that the SPBL also outperforms most

TABLE III: Comparison results of clinical skin disease diagnosis on the SD-198 dataset. SIFT and CN (color name) are extracted by using the code of [94]. "-ft" means fine-tuning the VggNet on SD-198. TS-L is Texture Symmetry of Lesion; CN-L is Color Name of Lesion; AC-L is Adaptive Compactness of Lesion; 'General-D" is the recognition accuracy of the general doctor who does not focus on one specific kind of disease; "Junior-D" is the recognition accuracy of junior dermatologist; C-Int is the intergeneration of three kinds of representations TS-L, CN-L and AC-L.

Method	SIFT [94]	CN [94]	Vgg [4]	Vgg-ft [4]	TS-L [74]	CN-L [74]	AC-L [74]	G-Doctor [74]	J-Doctor [74]	C-Int [74]	**Ours**
Acc	32.1±4.9	25.3±4.2	39.5±2.3	56.9±1.6	52.0±3.6	43.1±3.1	42.4±4.0	49.0	52.0	59.4±2.1	**67.8**±1.8

TABLE IV: Comparison with the state-of-the-art imbalanced learning methods on the tasks of scene classification (MIT-67 [80]), object classification (Caltech-101 [81]), handwritten digit classification (MNIST [82]) and coral classification (MLC [83]). We randomly set 50 sampling lists of the first three datasets respectively and report the mean performance since we can not get the list in the original paper, except for the MLC.

Dataset	SMOTE [47]	RUS [88]	SMOTE-RSB* [95]	WSVM [96]	WRF [97]	SOSR CNN [26]	CoSen CNN [27]	Rescale$_{new}$ [17]	**Ours**
MIT-67	33.9	28.4	34.0	35.5	35.2	49.8	56.9	35.1±1.2	**64.1**±0.5
Caltech-101	67.7	61.4	68.2	70.1	68.7	77.4	83.2	58.1±0.7	**88.6**±0.4
MNIST	94.5	92.1	96.0	96.8	96.3	97.8	98.6	98.1±0.3	**99.0**±0.1
MLC	38.9	31.4	43.0	47.7	46.5	65.7	68.6	63.7	**72.0**

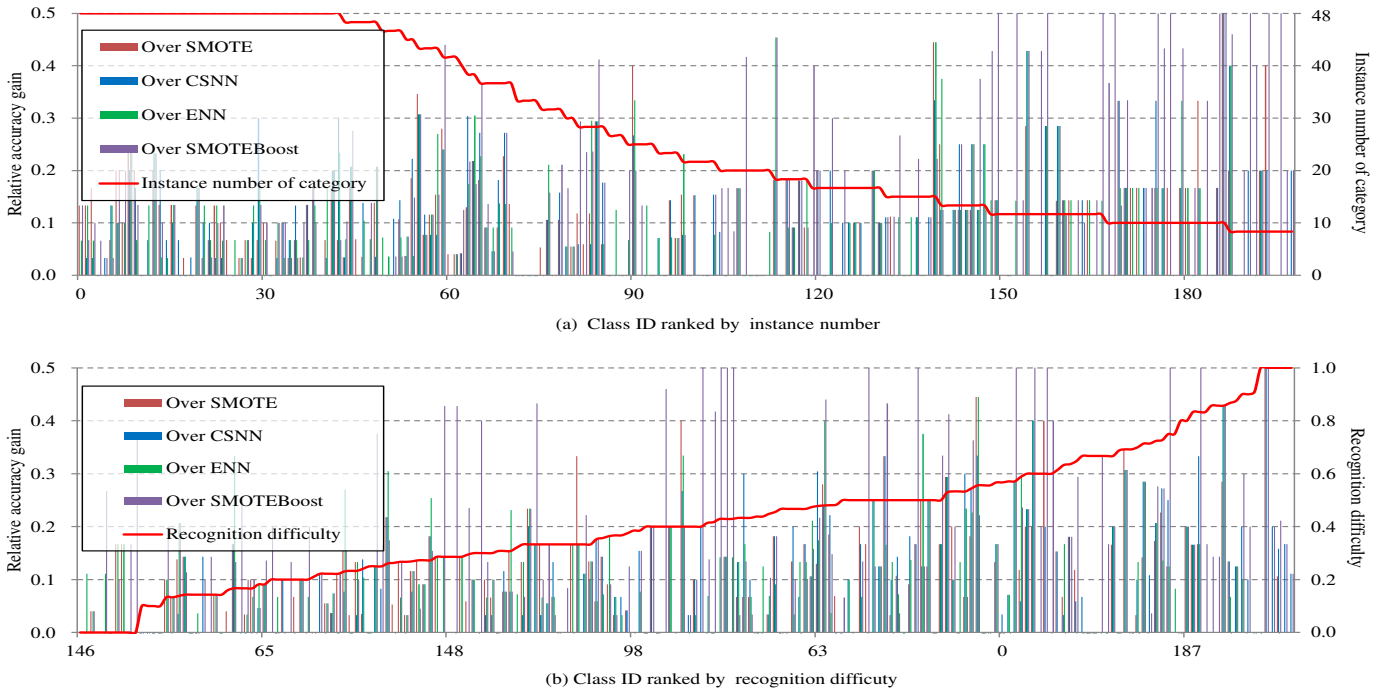
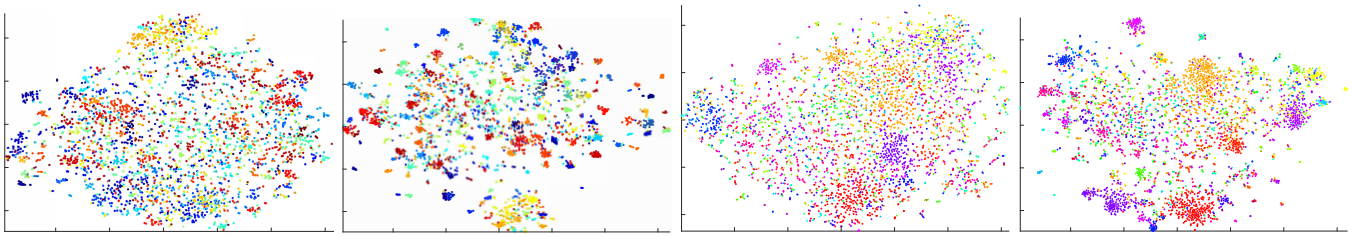


Fig. 5: Accuracy gains of SPBL over the comparative methods on the SD-198 dataset. (a) The class IDs are ranked by the instance number of categories from large to small, which are drawn by the red line. (b) The class IDs are ranked by the recognition difficulty (calculated by Eq. 6) of categories from easy to difficult, which are drawn by the red line. For both sub-figures, Y-axis (left) indicates the accuracy gains of SPBL against the other four methods. Y-axis of (a) (top right) refers to instance number of each class. Y-axis of (b) (bottom right) is the recognition difficulty of each class.

of the cost-sensitive based methods.

The CSNN method does not perform well which only get little improvement over the baseline. Its poor performance is also reflected in the G-mean value. The CSNN performs well on the binary classification task while faces more difficulty on the multi-class [48]. This shows that cost-sensitive learning is difficult with the increase in the number of classes in non-binary classification imbalance problems.

The ENN method performs the best except for the SPBL under the metrics "Precision", "Recall", "G-mean" and "F1" as shown in Table II. It even outperforms the SPBL in the G-mean metric by 2% on the SD-260 dataset, yet it achieves 9.9% lower performance of classification accuracy than our method. It efficiently measures the relative nearest neighbor



(a) (b) (c) (d)

Fig. 6: Visualizations of 2D t-SNE [98] feature embedding on the SD-198 (a-b) and SD-260 (c-d) datasets. (a) and (c) are the feature embedding using the features extracted from the raw ResNet50, *i.e.*, trained using all samples without the consideration of class imbalance and the SPL paradigm, on the SD-198 and SD-260 datasets, respectively. (b) and (d) are the feature embedding by using the features derived from the model of SPBL trained on SD-198 and SD-260, respectively. Note the models in all figures are trained with the same number of epochs.

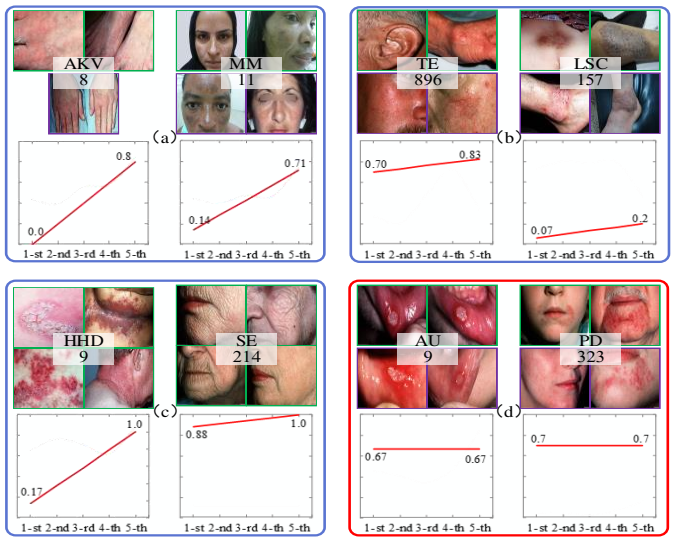


Fig. 7: Illustrations of classification results and the change of the recognition accuracy. The four sub-pictures are the positive examples (blue box) and negative examples (red box) of SPBL. For each sub-picture, the images with the green edge are correctly classified and the others are misclassified (purple). Each abbreviation above the image denotes the category of skin disease, *e.g.*, acrokeratosis verruciformis (AKV) and melasma (MM). The numbers below the abbreviations are the training instances number of the category. The red line denotes the change of classification accuracies of the 1-st and 5-th paces.

(NN) relationships among instances. This result indicates that it is important to define the relationships between instances or classes when designing the cost matrices, and it is not enough to only measure the class size in imbalance learning.

4) Comparison to Ensemble-based Methods: The SMOTEBoost and RUSBoost methods aim to improve the classification accuracy by integrating the decisions of several classifiers. For fair comparisons, we use the one-vs-rest SVMs with the same parameter settings as their base classifiers for these two comparison methods.

The ensemble-based methods we choose to compare perform similarly as the CSNN method, *i.e.*, outperforming the baseline method in most of the evaluation metrics but showing distinctly poor performance in the G-mean value. The RUSBoost method and the boosting learning procedure shows the positive effect in classification accuracy compared to the base RUS method but performs poorly especially in terms of G-mean on the level of the imbalance problem. The SMOTEBoost even performs poorly compared to the base SMOTE method on all evaluation metrics, although it slightly outperforms the baseline and the RUSBoost methods. The performance of SPBL demonstrates that the proposed method is capable of achieving good performance with a single classifier.

5) Comparison to Skin Disease Diagnosis Methods: We also compare the SPBL method with the state-of-the-art computer-aided diagnosis (CAD) methods in a clinical skin disease recognition task. The method proposed by Sun *et al.* [4] provided the SD-198 benchmark dataset and applied several state-of-the-art methods to it. For a fair comparison, we use the combination of the deep CNN features plus the SVM classifier as the method of this work to compare against, and it is noticeable that the results of this method exceed any results reported in [4]. The method proposed by Yang *et al.* [74] designed six medical representations considering different criteria for their diagnosis system. For the different experimental environment, *i.e.*, different training/testing split, we perform the five cross-validation experiment and report the average accuracy and standard deviation in Table III.

Both of the comparative methods especially the method proposed by Yang *et al.* [74] achieve comparable results with the dermatologists. However, there is a considerable number of methods in Table II, including the SPBL, that outperform them. When compared with [4] and [74] in terms of classification accuracy, the SPBL produces significant improvements of 10.9% and 8.4% respectively on the SD-198 dataset. The experimental results demonstrate the effectiveness of the SPBL and the validity of solving this real-world application with the imbalanced learning consideration.

6) Further Analysis: Fig. 5 shows the accuracy gains for each class of SPBL over the contrast methods, *i.e.*, SMOTE, CSNN, ENN and SMOTEBoost, on the SD-198 dataset. Our SPBL method solves the class imbalance issue based on both

the size and the recognition difficulty of each class. We show the improvement in two ways, *i.e.*, reordering the classes by class size and recognition difficulty respectively calculated by Eq. 6.

We can see that SPBL performs well on the classes with fewer instances and lower difficulty. Moreover, the SPBL shows a relatively balanced gain over competitors, *i.e.*, it improves the classification performance on classes no matter whether it is large or small, and is hard or easy. Traditional imbalanced learning methods mainly focus on the minority classes with a smaller size or higher complexity level. The proposed SPBL method considers all classes and aims to learn a balanced representation, as the results illustrated in Fig. 6, which outperforms the compared methods.

Fig. 7 visualizes several categories of clinical skin diseases and the change of recognition accuracies at different paces. Fig. 7 (a), (b) and (c) show that the SPBL performs well on both the categories with big or small sizes (such as "acrokeratosis verruciformis" (AKV), "melasma" (MM), "Telangiectasia" (TE), "lichen simplex chronicus" (LSC), "hailey-hailey disease" (HHD) and "solar elastosis" (SE)). The SPBL gradually learns the data from easy to hard, which can recognize the skin lesion that has a great change at the different stage of illness (*e.g.*, early and late stages). For example, the HHD in (c) has significantly different symptoms within-class in terms of border, color and lesion location at different stages, which can be gradually recognized by the proposed SPBL with only 6 training instances. As for the negative example of the results "aphthous ulcer" (AU) and "perioral dermatitis" (PD) of sub-picture (d), the recognition accuracies are not further improved during SPBL's learning, because the diagnosis of these diseases usually requires a biopsy. For example, distinguishing between AU and "Hand-foot-mouth Disease" often needs the liquid from the vesicula to be assayed, and the two skin diseases have very similar clinical manifestations. We also evaluate the proposed method on several other tasks as shown in Table IV, which also demonstrate the favorable performance of the SPBL against the comparative methods.

V. Conclusion

In this paper, we address the class imbalance issue and propose a novel SPBL algorithm which is trained using samples from easy to hard. We also propose a novel insight that in real-world applications, the class imbalance problem is not only due to the imbalanced distribution of class sizes but also the imbalanced recognition difficulty. Inspired by that, we propose both the penalty weight updating and curriculum reconstruction strategies which ensure that the model learns a comprehensively balanced representation in each self-paced learning procedure. We conduct experiments on two imbalanced datasets about clinical skin disease recognition tasks and several other imbalanced problems. The results indicate that both components of the proposed algorithm are effective and demonstrate the advantage of the SPBL against the state-of-the-art methods.

Acknowledgment

This work was supported by the NSFC (NO.61876094), Natural Science Foundation of Tianjin, China (NO. 18JCYBJC15400, 18ZXZNGX00110), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), and the Fundamental Research Funds for the Central Universities.

References

[1] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016. 1

[2] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016. 1

[3] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance," *Neural Networks*, vol. 21, no. 2-3, pp. 427–436, 2008. 1

[4] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," in *European Conference on Computer Vision*, 2016, pp. 206–222. 1, 2, 4, 7, 10, 11, 12

[5] W. Stolz, A. Riemann, A. Cognetta, L. Pillet, W. Abmayr, D. Holzel, P. Bilek, F. Nachbar, and M. Landthaler, "ABCD rule of dermatoscopy-a new practical method for early recognition of malignant-melanoma," *European Journal of Dermatology*, vol. 4, no. 7, 1994. 1

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 1, 7, 8

[7] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, and Z. Wei, "A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4152–4165, 2018. 1, 3, 8

[8] C. Huang, C. C. Loy, and X. Tang, "Discriminative sparse neighbor approximation for imbalanced learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1503–1513, 2018. 1, 3

[9] S. Chen, H. He, and E. A. Garcia, "RAMOBoost: Ranked minority oversampling in boosting," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1624–1642, 2010. 1, 3

[10] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *European Conference on Machine Learning*, 2004, pp. 39–50. 1

[11] J.-H. Xue and D. M. Titterington, "Do unbalanced data have a negative effect on LDA?" *Pattern Recognition*, vol. 41, no. 5, pp. 1558–1571, 2008. 1

[12] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 40–49, 2004. 1

[13] W. Elazmeh, N. Japkowicz, and S. Matwin, "Evaluating misclassifications in imbalanced data," in *European Conference on Machine Learning*, 2006, pp. 126–137. 1

[14] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *IEEE International Joint Conference on Neural Networks*, 2008, pp. 1322–1328. 1, 9, 10

[15] A. Nickerson, N. Japkowicz, and E. E. Milios, "Using unsupervised learning to guide resampling in imbalanced data sets," in *International Conference on Artificial Intelligence and Statistics*, 2001, pp. 261–265. 1

[16] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004. 1

[17] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," in *AAAI Conference on Artificial Intelligence*, 2006, pp. 567–572. 1, 3, 9, 10, 11

[18] X. Wang, X. Liu, N. Japkowicz, and S. Matwin, "Resampling and cost-sensitive methods for imbalanced multi-instance learning," in *International Conference on Data Mining Workshops*, 2013, pp. 808–816. 1

[19] X. Wang, S. Matwin, N. Japkowicz, and X. Liu, "Cost-sensitive boosting algorithms for imbalanced multi-instance datasets," in *Canadian Conference on Artificial Intelligence*, 2013, pp. 174–186. 1

[20] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and Information Systems*, vol. 25, no. 1, pp. 1–20, 2010. 2

[21] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999. 2

[22] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, 2010, pp. 1189–1197. 2, 3, 5, 8

[23] D. Meng, Q. Zhao, and L. Jiang, "A theoretical understanding of self-paced learning," *Information Sciences*, vol. 414, pp. 319–328, 2017. 2, 3, 4, 6, 8

[24] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, "Decorrelation of neutral vector variables: Theory and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 129–143, 2018. 2

[25] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational bayesian matrix factorization for bounded support data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 876–889, 2015. 2

[26] Y.-A. Chung, H.-T. Lin, and S.-W. Yang, "Cost-aware pre-training for multiclass cost-sensitive deep learning," *arXiv preprint arXiv:1511.09337*, 2015. 2, 11

[27] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3573–3587, 2018. 2, 3, 7, 11

[28] J. Yang, X. Sun, Y.-K. Lai, L. Zheng, and M.-M. Cheng, "Recognition from web data: A progressive filtering approach," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5303–5315, 2018. 2

[29] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3171–3183, 2017. 2

[30] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2473–2483, 2018. 2

[31] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *International Journal of Computer Vision*, vol. 127, no. 4, pp. 363–380, 2019. 2

[32] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265–278, 2019. 2

[33] Z. Ma, Y. Lai, W. B. Kleijn, Y.-Z. Song, L. Wang, and J. Guo, "Variational bayesian learning for dirichlet process mixture of inverted dirichlet distributions in non-Gaussian image feature modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 449–463, 2018. 2

[34] Z. Ma, H. Yu, W. Chen, and J. Guo, "Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 121–128, 2019. 2

[35] Z. Ma, J. Xie, Y. Lai, J. Taghia, J.-H. Xue, and J. Guo, "Insights into multiple/single lower bound approximation for extended variational inference in non-Gaussian structured data modeling," *IEEE Transactions on Neural Networks and Learning Systems*, 2019. 2

[36] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. 2, 3

[37] V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6585–6608, 2012. 2

[38] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013. 2

[39] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys*, vol. 49, no. 2, p. 31, 2016. 2

[40] T. Maciejewski and J. Stefanowski, "Local neighbourhood extension of SMOTE for mining imbalanced data," in *IEEE Symposium on Computational Intelligence and Data Mining*, 2011, pp. 104–111. 3

[41] B. Tang and H. He, "KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning," in *IEEE Congress on Evolutionary Computation*, 2015, pp. 664–671. 3

[42] Q. Dong, S. Gong, and X. Zhu, "Class rectification hard mining for imbalanced deep learning," in *IEEE International Conference on Computer Vision*, 2017, pp. 1851–1860. 3

[43] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, "Classification of imbalanced data by oversampling in kernel space of support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4065–4076, 2018. 3

[44] C. Drummond, R. C. Holte *et al.*, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *ICML Workshop on Learning from Imbalanced Datasets*, vol. 11, 2003, pp. 1–8. 3

[45] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Man*, vol. 39, no. 2, pp. 539–550, 2009. 3

[46] S. Chen and H. He, "Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach," *Evolving Systems*, vol. 2, no. 1, pp. 35–50, 2011. 3

[47] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. 3, 9, 10, 11

[48] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006. 3, 10, 11

[49] C. L. Castro and A. P. Braga, "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 888–899, 2013. 3

[50] T.-K. Jan, D.-W. Wang, C.-H. Lin, and H.-T. Lin, "A simple methodology for soft cost-sensitive classification," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 141–149. 3

[51] N. Abe, B. Zadrozny, and J. Langford, "An iterative method for multi-class cost-sensitive learning," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 3–11. 3

[52] B. Mirza and Z. Lin, "Meta-cognitive online sequential extreme learning machine for imbalanced and concept-drifting data classification," *Neural Networks*, vol. 80, pp. 79–94, 2016. 3

[53] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, "AdaCost: Misclassification cost-sensitive boosting," in *International Conference on Machine Learning*, vol. 99, 1999, pp. 97–105. 3

[54] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999. 3

[55] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000. 3

[56] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. 3

[57] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Cybernetics*, vol. 42, no. 4, pp. 463–484, 2012. 3

[58] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach," *ACM Explorations Newsletter*, vol. 6, no. 1, pp. 30–39, 2004. 3

[59] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1356–1368, 2015. 3

[60] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *IEEE Symposium on Computational Intelligence and Data Mining*, 2009, pp. 324–331. 3

[61] J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," *Neurocomputing*, vol. 150, pp. 529–542, 2015. 3

[62] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *International Conference on Machine Learning*, 2000, pp. 983–990. 3

[63] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007. 3

[64] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Advances in Neural Information Processing Systems*, 2017, pp. 7032–7042. 3

[65] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2160–2173, 2011. 3

[66] H. Yu, Z.-H. Tan, Z. Ma, R. Martin, and J. Guo, "Spoofing detection in automatic speaker verification systems using dnn classifiers and dynamic acoustic features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4633–4644, 2018. 3

[67] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Annual International Conference on Machine Learning*, 2009, pp. 41–48. 3

[68] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *ACM International Conference on Multimedia*, 2014, pp. 547–556. 3, 5

[69] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for matrix factorization," in *AAAI Conference on Artificial Intelligence*, 2015, pp. 3196–3202. 3, 5

[70] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *AAAI Conference on Artificial Intelligence*, 2015, pp. 2694–2700. 3

[71] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 865–878, 2017. 3

[72] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, "Active self-paced learning for cost-effective and progressive face identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 7–19, 2018. 3

[73] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017. 4

[74] J. Yang, X. Sun, L. Jie, and R. Paul, "Clinical skin lesion diagnosis using representations inspired by dermatologist criteria," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1258–1266. 4, 10, 11, 12

[75] J. Kawahara, A. BenTaieb, and G. Hamarneh, "Deep features to classify skin lesions," in *IEEE International Symposium on Biomedical Imaging*, 2016, pp. 1397–1400. 4

[76] L. Yu, H. Chen, Q. Dou, J. Qin, and P.-A. Heng, "Automated melanoma recognition in dermoscopy images via very deep residual networks," *IEEE Transactions on Medical Imaging*, vol. 36, no. 4, pp. 994–1004, 2017. 4

[77] M. Das Gupta, S. Srinivasa, M. Antony *et al.*, "KL divergence based agglomerative clustering for automated vitiligo grading," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2700–2709. 4

[78] D. Gutman, N. C. Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC)," *arXiv preprint arXiv:1605.01397*, 2016. 4

[79] K. M. Ting, "An instance-weighting method to induce cost-sensitive trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, pp. 659–665, 2002. 5

[80] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420. 7, 11

[81] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007. 7, 11

[82] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 7, 11

[83] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1170–1177. 7, 11

[84] A. Fernández, S. García, M. J. del Jesus, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets and Systems*, vol. 159, no. 18, pp. 2378–2398, 2008. 7

[85] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. 7

[86] Y. Sun, M. S. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalanced class distribution," in *International Conference on Data Mining*, 2006, pp. 592–602. 7

[87] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learning*, vol. 45, no. 2, pp. 171–186, 2001. 7

[88] I. Mani and I. Zhang, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *ICML Workshop on Learning from Imbalanced Datasets*, vol. 126, 2003. 9, 10, 11

[89] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Machine Learning*, vol. 95, no. 2, pp. 225–256, 2014. 9, 10

[90] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," *Advances in Intelligent Computing*, vol. 3644, pp. 878–887, 2005. 9, 10

[91] B. Tang and H. He, "ENN: Extended nearest neighbor method for pattern recognition," *IEEE Computational Intelligence Magazine*, vol. 10, no. 3, pp. 52–60, 2015. 10

[92] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," *Knowledge Discovery in Databases*, vol. 2838, pp. 107–119, 2003. 10

[93] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems*, vol. 40, no. 1, pp. 185–197, 2010. 10

[94] C. Göring, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2489–2496. 11

[95] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory," *Knowledge and Information Systems*, vol. 33, no. 2, pp. 245–265, 2012. 11

[96] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Transactions on Cybernetics*, vol. 39, no. 1, pp. 281–288, 2009. 11

[97] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, pp. 1–12, 2004. 11

[98] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008. 12